IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 19, 2019, accepted June 7, 2019, date of publication June 12, 2019, date of current version June 25, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2922199*

# Model Matching: A Novel Framework to use Clustering Strategy to Solve the Classification Problem

**ZHIYI DUAN[1], LIMIN WANG[1,2], AND MINGHUI SUN[1]**
[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China

Corresponding author: Minghui Sun (smh@jlu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61272209 and Grant 61872164.

**ABSTRACT** It is a common practice to handle labeled data with classifiers and unlabeled ones with clusterings. The traditional Bayesian network classifiers (BNC$^{\mathcal{T}}$s) learned from labeled training set $\mathcal{T}$ directly map the unlabeled test instance into the network structure to calculate the conditional probability for the classification, which neglects the information hidden in the unlabeled data and will result in classification bias. To address this issue, we propose a novel learning framework, called model matching, that uses the ''clustering'' strategy to solve the classification problem. The labeled data is divided into several clusters according to the different class label to learn a set of BNC$^{\mathcal{T}}$s and a corresponding set of BNC$^{p}$s is built for each unlabeled test instance. To make a classification, the cross entropy method is applied to compare the structural similarity between BNC$^{\mathcal{T}}$ and BNC$^{p}$. The extensive experimental results on 46 datasets from the University of California at Irvine (UCI) machine learning repository demonstrate that for BNCs model matching helps improve the generalization performance and outperforms the several state-of-the-art classifiers like tree-augmented naive Bayes and Random forest.

**INDEX TERMS** Bayesian network, model matching, unlabeled data, cross entropy.

## I. INTRODUCTION

Classification and clustering are both the basic tasks in data analysis and machine learning, which have attracted widespread attention in recent years. Given a set of $z$ labels of the class variable $C$, the task of supervised classification can be defined as the assignment of a label $c$ to an unlabeled test instance $\boldsymbol{x} = (x_1, \cdots, x_n)$, with the values for the $n$ attributes $\boldsymbol{X} = \{X_1, \cdots, X_n\}$. There are numerous classification techniques [1]–[7], among which Bayesian network classifiers (BNCs) have long been a popular tool for graphically representing the probabilistic dependencies which exist in a domain. Such BNCs can be learned in a lot of ways, and discriminative learning [8]–[10] directly models the conditional probability $P(c|\boldsymbol{x})$, which corresponds to optimizing an objective function that is highly representative of classification error, such as maximizing class conditional likelihood. Unfortunately, there have been a number of negative results over the past years, showing that discriminatively learning various forms of BNCs is NP hard [11], largely because the cost functions that are needed to be optimized do not in general decompose. To address this issue, by applying Bayes' theorem, generative learning [12]–[16] approximates the joint probability $P(c|\boldsymbol{x})$ according to BNCs with diverse factorizations and the classification process can be done in the following way:

$$\begin{aligned}\arg\max_{C} P(c|\boldsymbol{x}) &= \arg\max_{C} \frac{P(c,\boldsymbol{x})}{P(\boldsymbol{x})}\\ &\propto \arg\max_{C} P(c,\boldsymbol{x})\\ &= \arg\max_{C} P(c)P(\boldsymbol{x}|c). \qquad (1)\end{aligned}$$

Naive bayes (NB) [17], which is the simplest BNC, assumes that all the predictive attributes are independent of each other given the class variable. Although NB is surprisingly effective, as the amount of data increases, the dependencies between predictive attributes make the performance of NB degrade dramatically. Researchers have proposed a lot of prior work that has explored approaches to alleviate NB's independence assumption. Some restricted BNCs,

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Chen.

which suppose that each predictive attribute is directly dependent on the class variable $C$, extend the structure of NB by adding a bounded number of additional interdependencies [18]–[20], e.g., tree-augmented naive Bayes (TAN) [21] and $k$-dependence Bayesian classifier (KDB) [22]. At the other extreme, attribute weighting methods [23], [24] have been widely used as a means of increasing the influence of major predictive attributes. All the above mentioned BNCs learn from labeled data in training set $\mathcal{T}$, i.e., BNC$^{\mathcal{T}}$, but the dependencies hidden in unlabeled test instances have received relatively little attention. Scientific dataset can be massive and labeled training data may account for only a small portion. That is to say, BNC$^{\mathcal{T}}$ can only represent a limited number of conditional dependencies. Moreover, the structure of BNC$^{\mathcal{T}}$ is definite and can not adjust to diverse unlabeled test instances automatically, since the dependencies that exist in different unlabeled test instances may differ greatly.

Most restricted BNCs model the network structure by mining the dependence relationships between predictive attributes, which are usually measured by conditional mutual information (CMI). Given the class variable $C$, CMI between predictive attributes $X_i$ and $X_j$, or $I(X_i; X_j|C)$, can be calculated as follows [25],

$$\begin{aligned} I(X_i; X_j|C) &= \sum_{x_i}\sum_{x_j}\sum_{c} P(x_i, x_j, c) log\frac{P(x_i, x_j|c)}{P(x_i|c)P(x_j|c)} \\ &= \sum_{c} P(c)\sum_{x_i}\sum_{x_j} P(x_i, x_j|c) log\frac{P(x_i, x_j|c)}{P(x_i|c)P(x_j|c)} \\ &= \sum_{c} P(c)\hat{I}(X_i; X_j|c) \\ &= \sum_{x_i}\sum_{x_j}\sum_{c} \hat{I}(x_i; x_j|c) \end{aligned} \tag{2}$$

However, for unlabeled instances, because of their uncertain class labels it is unsuitable to use CMI to measure the dependence relationships between attribute values, which may result in classification bias. Clustering [26]–[28] is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups, which is one common practice to handle unlabeled instances. Model-based clustering combines classification and clustering strategies and defines a cluster as a component in a mixture model, which has received much attention [29]–[31]. To mine the dependence relationships hidden in unlabeled test instance with respect to different possible class labels, in this paper we propose a novel learning framework, called model matching, which first divides the training set into $z$ clusters according to different class labels, where $z$ is the number of class labels. For each cluster, a general Bayesian model BNC$^{\mathcal{T}}$ is built. Corresponding to the unlabeled test instance $\boldsymbol{x} = (x_1, \cdots, x_n)$, by pre-assigning class label to it we can build a "pseudo" training set $\mathcal{P}$ where $\mathcal{P} = \{(\boldsymbol{x}, c_1), (\boldsymbol{x}, c_2), \cdots, (\boldsymbol{x}, c_z)\}$. For each "pseudo" instance in $\mathcal{P}$, a specific Bayesian model BNC$^{\mathcal{P}}$ is built independently in classification phase. Based on "clustering" method, the cross entropy is applied to compare the structure similarity of BNC$^{\mathcal{T}}$ and BNC$^{\mathcal{P}}$ to make the final prediction. Through extensive experiments on 46 UCI (University of California at Irvine) datasets, we prove that the model matching framework can alleviate the potential misclassification problem without causing too many offsetting errors or incurring very high computation overhead.

The rest of this paper is organized as follows. Section II introduces some state-of-the-art restricted BNCs. Section III explains the basic idea of the model matching in detail. Section IV compares experimental results on datasets from the UCI Machine Learning Repository. Section V draws conclusion.

## II. RESTRICTED BAYESIAN NETWORK CLASSIFIERS

The restricted BNCs [32] are a way to graphically represent the dependencies in a probability distribution $\Theta$ by the construction of a directed acyclic graph $\mathcal{G}$. Nodes in $\mathcal{G}$ represent the attributes $\boldsymbol{X}$ or class variable $C$, and arcs denote the probability dependencies between child nodes and their parent nodes. Parameter $\Theta$ uses conditional probability to quantitatively describe the conditional dependencies for each node in $\mathcal{G}$, namely $P(x_i|\Pi_i, c)$, where $\Pi_i$ is a set of parent attributes of $X_i$. Nodes with no parents simply represent the prior probability for that attribute. As shown in Figure 1a, the full Bayesian network classifier (FBC) [33] completely reflects the dependencies between predictive attributes and thus will achieve optimal performance. By using the chain rule of joint probability distribution, $P(c, \boldsymbol{x})$ can be calculated as follows:

$$P(c, \boldsymbol{x}) = P(c)\prod_{i=1}^{n} P_{FBC}(x_i|c, \Pi_i), \tag{3}$$

where $\Pi_i = \{X_1, \cdots, X_{i-1}\}$. However, it is very time consuming to learn a FBC, since the computational complexity grows exponentially until it becomes NP hard with the increasing number of attributes and arcs in the structure. To simplify the network structure, researchers have proposed numerous state-of-the-art classification algorithms [34]–[36].

As shown in Figure 1b, NB represents the most restrictive extreme in the spectrum of probabilistic classification techniques, which assumes that all the predictive attributes are conditionally independent given the class variable $C$. Although the conditional independence assumption rarely holds in the real world, NB has achieved competitive classification performance, especially when the data quantity is small [37].

TAN is an extension of NB, which imposes a tree structure to alleviate the NB's conditional independence assumption. As can be seen in Figure 1c, each predictive attributes in TAN has the class variable and at most one other attribute as parents. TAN constructs maximum weighted spanning tree (MWST) to represent dependence relationships between predictive attributes, which are measured by CMI. The learning procedure of TAN is depicted in Algorithm 1.

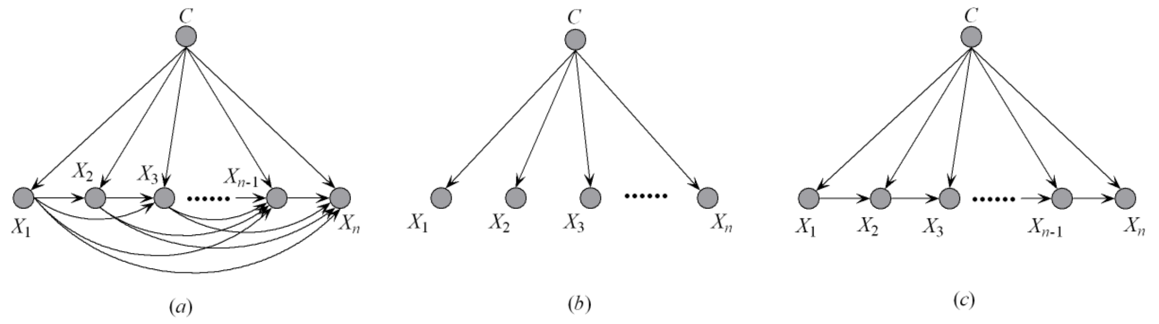


**FIGURE 1.** Example of (a) full Bayesian network classifier, (b) Naive Bayes and (c) Tree augmented naive Bayes.

**Algorithm 1** The TAN Learning Procedure

```
Input: Training set $\mathcal{T}$, attribute set $\{X_1, \cdots, X_n\}$ and class
    variable C.
Output: TAN classifier.
  1. Calculate $I(X_i; X_j|C)$ $(i \neq j)$ between each pair of
     attributes by Equation 2.
  2. Build a complete undirected graph in which the nodes
     are the attributes. Annotate the weight of an arc
     connecting $X_i$ to $X_j$ by $I(X_i; X_j|C)$ $(i \neq j)$.
  3. Build the maximum weighting spanning tree.
  4. Transform the resulting undirected tree to a directed
     one by choosing a root variable and setting the
     direction of all arcs to be outward from it.
  5. Construct a TAN classifier by adding a class variable
     $C$ and adding an arc from $C$ to each $X_i$.
  6. Return TAN classifier.
```

Although TAN is always regarded as the optimal one-dependence classifier [38], there is still a lot of prior work that has explored approaches to improve the classification performance of TAN. One approach is to optimize the structure of TAN by using other evaluation criterion. Ruz and Pham [39] proposes a method that a Bayesian criterion controls the likelihood of the data and the complexity of the TAN structure, which results in a predictor subgraph that minimizes the Kullback-Leibler (KL) divergence between the real joint probability distribution and the approximation given by the model. For handling with the imprecise probabilities estimation, Zaffalon and Fagiuoli [40] proposes the tree-based credal classifier algorithm which can induce credal Bayesian networks with a TAN structure. At the other extreme, bagging-type metaclassifiers use bootstrap samples and generate diverse results from the different classifiers. The bagging randomTAN [41] takes randomTAN, which randomly selects the arcs between predictive attributes whose CMI surpasses a fixed threshold, as base classifiers in a bagging scheme. The averaged TAN (ATAN) [36] takes each predictive attribute as a root node and then builds a set of TANs. In ATAN, the posterior probabilities produced by different TAN classifiers are directly averaged to make a prediction.

To choose suitable distributions for the probabilities, given the training data by $\mathcal{T} = (\hat{x}_{i1}, \cdots, \hat{x}_{in}, \hat{c}_i)$, $i \in \{1, \cdots, N\}$, where $N$ is the number of training instances, $\hat{c}_i \in \{c_1, \cdots, c_z\}$ is the class label of the $i$-th training instance, i.e., there are $z$ class labels, the prior and joint probabilities in the above equations will be calculated as follows,

$$\begin{cases} P(c) = \dfrac{1}{N}\displaystyle\sum_{k=1}^{N} \delta(c, \hat{c}_k) \\ P(x_j) = \dfrac{1}{N}\displaystyle\sum_{k=1}^{N} \delta(x_j, \hat{x}_{kj}) \\ P(x_j, c) = \dfrac{1}{N}\displaystyle\sum_{k=1}^{N} \delta(\langle x_j, c\rangle, \langle \hat{x}_{kj}, \hat{c}_k\rangle) \\ P(x_i, x_j, c) = \dfrac{1}{N}\displaystyle\sum_{k=1}^{N} \delta(\langle x_i, x_j, c\rangle, \langle \hat{x}_{ki}, \hat{x}_{kj}, \hat{c}_k\rangle) \end{cases} \tag{4}$$

where $\delta(\cdot)$ is a binary function, which is zero if its two parameters are different and one otherwise. $< \cdot >$ denotes the combination of attribute values. Then, $P(x_j|c)$ and $P(x_i, x_j|c)$ can be calculated as follows,

$$\begin{cases} P(x_j|c) = \dfrac{P(x_j, c)}{P(c)} \\ P(x_i, x_j|c) = \dfrac{P(x_i, x_j, c)}{P(c)} \end{cases} \tag{5}$$

The joint conditional probability, $P(x_i|c, \Pi_i)$, in Equation 3 can reflect the probabilistic dependence between attribute values given different class label $c$. However, from the viewpoint of information theory, CMI in Equation 2 can only measure the conditional dependence implicated in training set between predictive attributes given the class variable $C$, whereas cannot weigh the probabilistic dependence conditioned on the specific class label $c$. We argue that the difference between conditional dependence and probabilistic dependence may result in classification bias. In the following discussion, we explain the basic idea of model matching by using TAN as the base classifier and extend CMI to address this issue.

## III. THE MODEL MATCHING FRAMEWORK

As Figure 1 shows, these classic BNCs mentioned above, like FBC and TAN, which learn from training data $\mathcal{T}$ and apply different strategies to build the network structure, may represent different conditional dependencies between attributes. However, most of these strategies model the structure by describing the conditional dependencies between predictive attributes given the class variable $C$. Actually, from Equation 2, we can find that the value of $\hat{I}(X_i;X_j|c)$ may differ greatly for different class labels and CMI $= \sum_c P(c)\hat{I}(X_i;X_j|c)$. Thus, it may be unsuitable to use CMI to measure the dependence relationships between attributes $X_i$ and $X_j$ given specific class label $c$. This may result in the classification bias for BNCs, and we argue that it is a viable solution to model a set of network structures by mining the conditional dependencies between attributes given different class labels.

Clustering is the organization of a collection of unlabeled instances into clusters based on similarity [42]. The clustering problem has been addressed in many contexts and by researchers in many disciplines. Inspired by its success, the "clustering" strategy is applied in our algorithm to solve the classification problem and we first divide the training set into $z$ subsets according to different class labels, where $z$ is the number of class labels. To measure the dependence relationships existed in different subsets, the label based CMI is proposed and defined as follows,

*Definition 1:* The label based CMI (LCMI) is defined to measure the amount of information shared between two predictive attributes $X_i$ and $X_j$ given a specific class label $c$; that is,

$$I(X_i;X_j|c) = \sum_{x_i}\sum_{x_j}\hat{P}(x_i,x_j|c)log\frac{\hat{P}(x_i,x_j|c)}{\hat{P}(x_i|c)\hat{P}(x_j|c)}. \qquad (6)$$

The Equation 6 is derived from the second step of Equation 2. For different training subsets, the dependence relationships in TAN will be measured by LCMI instead of CMI for model construction. We refer to a set of improved TAN as $\text{TAN}^{\mathcal{T}}_{\hat{z}}$, where $\hat{z} \in \{1, \cdots, z\}$.

Given a particular unlabeled test instance $p = (x_1, \cdots, x_n, C = ?)$, from the viewpoint of clustering, we need to consider the clusters which it belongs to. To assign the class label to $p$, only a small number of dependencies, which are set in $\text{TAN}^{\mathcal{T}}$, are necessary. The other dependencies in $\text{TAN}^{\mathcal{T}}$ may counteract the effect of the necessary dependencies. The proposed approach aims to give high priority the dependencies that related to the elements in $p$. Corresponding to $\text{TAN}^{\mathcal{T}}_{\hat{z}}$, a set of $\text{TAN}^{p}_{\hat{z}}$ is built independently. To describe the probabilistic dependence existed in $p$, instance-based CMI is proposed and defined as follow,

*Definition 2:* The instance-based CMI (ICMI) is defined to measure the amount of information shared between two predictive attribute values $x_i$ and $x_j$ given a specific class label $c$; that is,

$$I(x_i;x_j|c) = \hat{P}(x_i,x_j|c)log\frac{\hat{P}(x_i,x_j|c)}{\hat{P}(x_i|c)\hat{P}(x_j|c)}. \qquad (7)$$

where

$$\begin{cases} \hat{P}(c) = \dfrac{1}{N+1}[\sum_{k=1}^{N}\delta(c,\hat{c}_k) + \dfrac{1}{z+1}] \\ \hat{P}(x_j) = \dfrac{1}{N+1}[\sum_{k=1}^{N}\delta(x_j,\hat{x}_{kj}) + 1] \\ \hat{P}(x_j,c) = \dfrac{1}{N+1}[\sum_{k=1}^{N}\delta(\langle x_j,c\rangle, \langle\hat{x}_{kj},\hat{c}_k\rangle) + \dfrac{1}{z+1}] \\ \hat{P}(x_i,x_j,c) = \dfrac{1}{N+1}[\sum_{k=1}^{N}\delta(\langle x_i,x_j,c\rangle, \langle\hat{x}_{ki},\hat{x}_{kj},\hat{c}_k\rangle) \\ \qquad + \dfrac{1}{z+1}] \end{cases} \qquad (8)$$

Conditional probability can be calculated as follows,

$$\begin{cases} \hat{P}(x_j|c) = \dfrac{\hat{P}(x_j,c)}{\hat{P}(c)} \\ \hat{P}(x_i,x_j|c) = \dfrac{\hat{P}(x_i,x_j,c)}{\hat{P}(c)} \end{cases} \qquad (9)$$

Similar to the Laplace correction [43], the main idea behind Equation 8 is equivalent to creating a "pseudo" training set $\mathcal{P}$ by adding to the training set $\mathcal{T}$ a new instance $(x_1, \cdots, x_n)$ with multi-label by assuming that the probability that this new instance is in class $c$ is $1/z$ for each $c \in \{c_1, \cdots, c_z\}$. To make a fair comparison of the structure of $\text{TAN}^{\mathcal{T}}$ and $\text{TAN}^{p}$, the same learning strategy is applied to build them. The learning procedures of $\text{TAN}^{\mathcal{T}}_{\hat{z}}$ and corresponding $\text{TAN}^{p}_{\hat{z}}$ are depicted in algorithms 2 and 3, respectively.

**Algorithm 2** The $\text{TAN}^{\mathcal{T}}$ Learning Procedure

```
Input: Training set $\mathcal{T}$, attribute set $\{X_1, \cdots, X_n\}$ and class
    label $\{c_1, \cdots, c_z\}$.
Output: a set of $\text{TAN}^{\mathcal{T}}_{\hat{z}}$ classifier.
Let $\hat{z} = 1$, for $\hat{z} \le z$:
  1. Calculate $I(X_i;X_j|c_{\hat{z}})$ $(i \neq j)$ between each pair of
     attributes by Equation 6.
  2. Build a complete undirected graph in which the nodes
     are the attributes. Annotate the weight of an arc
     connecting $X_i$ to $X_j$ by $I(X_i;X_j|c_{\hat{z}})$ $(i \neq j)$.
  3. Build the maximum weighting spanning tree.
  4. Transform the resulting undirected tree to a directed
     one by choosing a root variable and setting the
     direction of all arcs to be outward from it.
  5. Construct a $\text{TAN}^{\mathcal{T}}_{\hat{z}}$ classifier by adding a class label $c_{\hat{z}}$
     and adding an arc from $c_{\hat{z}}$ to each $X_i$.
  6. Return a set of $\text{TAN}^{\mathcal{T}}_{\hat{z}}$ classifier.
```

Then, after training the classifiers, traditional BNCs usually map the attribute values into the network structure to calculate joint probability for classification. It is based on the fact that for one single BNC the value of $P(\boldsymbol{x})$ in Equation 1 is the same. However, for different BNCs, the value of $P(\boldsymbol{x})$ may differ greatly and this leads to the joint probability

**Algorithm 3** The TAN$^p$ Learning Procedure

```
Input: Training set $\mathcal{T}$ and test instance $p$.
Output: a set of $TAN^{p}_{\hat{z}}$ classifier.
Let $\hat{z} = 1$, for $\hat{z} \leq z$:
  1. Calculate $I(x_i; x_j|c_{\hat{z}})$ $(i \neq j)$ between each pair of
     attribute values by Equation 7.
  2. Build a complete undirected graph in which the nodes
     are the attribute values. Annotate the weight of an arc
     connecting $x_i$ to $x_j$ by $I(x_i; x_j|c_{\hat{z}})$ $(i \neq j)$.
  3. Build the maximum weighting spanning tree.
  4. Transform the resulting undirected tree to a directed one
     by choosing a root attribute value and setting the
     direction of all arcs to be outward from it.
  5. Construct a $TAN^{p}_{\hat{z}}$ classifier by adding a class label
     $c_{\hat{z}}$ and adding an arc from $c_{\hat{z}}$ to each $x_i$.
  6. Return a set of $TAN^{p}_{\hat{z}}$ classifier.
```

in Equation 1 cannot be directly compared for classification. Thus, since we get a set of TAN$^{\mathcal{T}}_{\hat{z}}$s, the traditional estimation method cannot apply to our algorithm and may result in classification bias. Inspired by the clustering method, we address this issue by comparing the structure similarity between TAN$^{\mathcal{T}}$ and TAN$^p$. The cross entropy method [44] is a new generic approach to combinatorial and multi-extremal optimization and rare event simulation. In this paper, the cross entropy is applied to penalize the deviation between the actual output and the expected output, and is calculated by

$$H(q, \hat{q}) = -\sum_{x} q(x) log\, \hat{q}(x), \tag{10}$$

where both $q$ and $\hat{q}$ represent a set of probabilities. Since TAN$^{\mathcal{T}}$ is learned from the labeled training set, we consider that TAN$^{\mathcal{T}}$ can reflect higher degree of conditional dependencies and represent the actual probability distribution, while TAN$^p$ is built for each unlabeled test instance and can represent the expected output. Thus, in our algorithm, the conditional probabilities calculated by TAN$^{\mathcal{T}}_{\hat{z}}$ and TAN$^{p}_{\hat{z}}$ are considered as $q$ and $\hat{q}$, respectively. The TAN$^{\mathcal{T}}_{\hat{z}}$ with the minimum value of cross entropy is selected to make the final prediction.

For learning the network structure of TAN$^{\mathcal{T}}$, it requires $\mathcal{O}(n^2zNv^2)$ time that is dominated by the computations of LCMI, where $n$ is the number of predictive attributes, $N$ is the number of instances, $z$ is the number of class labels, and $v$ is the maximum number of discrete values that an attribute may take. For building the corresponding TAN$^p$, it takes only $\mathcal{O}(n^2zN)$ time which is dominated by the computations of ICMI. The small computational complexity makes the model matching framework very suitable for data mining domains.

## IV. EMPIRICAL STUDY

To illustrate the effectiveness of our proposed model matching framework, we conduct experiments on 46 datasets from the UCI machine learning repository [45]. The structure of the experimental Section is as follows: To begin with, Section 4.1 compares our TAN$^m$ (TAN that performs the model matching framework) algorithm with two state-of-the-art machine learning algorithms, Random forest [46] and Logistic regression [47]. Section 4.2 includes comparisons with three classic BNCs, that are NB, TAN and KDB. Section 4.3 presents a global comparison of all learners considered by applying the Friedman and Nemenyi tests. When two learners are compared, Win/Draw/Loss (W/D/L) record is applied to count the number of datasets for which one algorithm performs better, equally well or worse than the other on a given measure. We consider there exists a significant difference if the output of a one-tailed binomial sign test is less than 0.05. The detailed characteristics of each dataset are shown in Table 1 in ascending order of their sizes. These datasets are categorized in terms of their sizes. That is, datasets with instances $< 1000$, $\geq 1000$ and $< 10000$, $\geq 10000$ are denoted as small size, medium size and large

**TABLE 1.** Datasets.

| No. | Dataset | Instance | Attribute | Class |
|---|---|---|---|---|
| 1 | Lung-cancer | 32 | 56 | 3 |
| 2 | Labor | 57 | 16 | 2 |
| 3 | Promoters | 106 | 57 | 2 |
| 4 | Lymphography | 148 | 18 | 4 |
| 5 | Iris | 150 | 4 | 3 |
| 6 | Teaching-ae | 151 | 5 | 3 |
| 7 | Wine | 178 | 13 | 3 |
| 8 | Sonar | 208 | 61 | 2 |
| 9 | Glass-id | 214 | 9 | 3 |
| 10 | New-thyroid | 215 | 5 | 3 |
| 11 | Audio | 226 | 69 | 24 |
| 12 | Heart | 270 | 13 | 2 |
| 13 | Hungarian | 294 | 14 | 2 |
| 14 | Soybean-large | 307 | 35 | 19 |
| 15 | Bupa | 345 | 6 | 2 |
| 16 | Dermatology | 366 | 34 | 6 |
| 17 | Horse-colic | 368 | 21 | 2 |
| 18 | House-votes-84 | 435 | 17 | 2 |
| 19 | Cylinder-bands | 540 | 39 | 2 |
| 20 | Chess | 551 | 40 | 2 |
| 21 | Credit-a | 690 | 15 | 2 |
| 22 | Crx | 690 | 15 | 2 |
| 23 | Breast-cancer-w | 699 | 9 | 2 |
| 24 | Vehicle | 846 | 18 | 4 |
| 25 | German | 1000 | 20 | 2 |
| 26 | Segment | 2310 | 19 | 7 |
| 27 | Hypothyroid | 3163 | 25 | 2 |
| 28 | Splice-c4.5 | 3177 | 60 | 3 |
| 29 | Kr-vs-kp | 3196 | 36 | 2 |
| 30 | Dis | 3772 | 29 | 2 |
| 31 | Hypo | 3772 | 29 | 4 |
| 32 | Sick | 3772 | 29 | 2 |
| 33 | Spambase | 4601 | 57 | 2 |
| 34 | Waveform-5000 | 5000 | 41 | 3 |
| 35 | Phoneme | 5438 | 8 | 52 |
| 36 | Wall-following | 5456 | 24 | 4 |
| 37 | Page-blocks | 5473 | 10 | 5 |
| 38 | Optdigits | 5620 | 64 | 10 |
| 39 | Thyroid | 9169 | 29 | 20 |
| 40 | Pendigits | 10992 | 16 | 10 |
| 41 | Magic | 19020 | 10 | 2 |
| 42 | Letter-recog | 20000 | 16 | 26 |
| 43 | Adult | 48842 | 14 | 2 |
| 44 | Shuttle | 58000 | 9 | 7 |
| 45 | Localization | 164860 | 5 | 11 |
| 46 | Poker-hand | 1025010 | 10 | 10 |

size, respectively. We will report results on these sets to discuss suitability of a classifier for datasets of different sizes. For each dataset, we use MDL (Minimum Description Length) discretization [48] to discretize numeric attributes. Missing values are regarded as a distinct value. Each algorithm is tested on each dataset using 10-fold cross validation. The base probabilities of each algorithm are estimated using $m$-estimation ($m = 1$), since some researchers report that the $m$-estimation leads to more accurate probabilities than the Laplace estimation [49].

### A. TAN$^M$ VS TWO STATE-OF-THE-ART LEARNERS

In order to know how much predictive capacity is get promoted by using our model matching framework, it is useful to compare TAN$^m$ to the state-of-the-art learners, e.g., Random forest and Logistic regression.

#### 1) RANDOM FOREST

Random forest (RF) is considered as one of the most powerful learning algorithm [46]. It uses bagging methods to aggregate multiple decision trees that are trained on data selected at random but with replacement from the original data. Each decision tree is grown to its largest possible size and no pruning is done. When dealing with medium and large size datasets, RF containing 100 trees (RF100) is used in our experiments. Moreover, note that RF with 10 trees (RF10) is applied to handle small datasets, since the 100 decision trees are complex and tend to overfit the training data. To compare the classification accuracy of RF and TAN$^m$, zero-one loss function, which is the most common loss function to measure the misclassification rate, is applied in our experiments. The detailed results for each dataset in terms of zero-one loss can be found in Table 2. Table 3 presents the W/D/L records TAN$^m$ and RF in terms of zero-one loss. Note, it is unable to get results for RF on our largest dataset `Poker-hand` due to main memory constrains. Thus, this dataset will be removed in the following discussion in this section. Table 3 shows the W/D/L records when compared with RF. The results show that TAN$^m$ achieves better classification performance than RF on 21 datasets out of 45 datasets, providing solid evidence for the effectiveness of our proposed model matching framework.

To further show the performance of TAN$^m$ when compared with RF over datasets of diverse size, the goal difference (GD) function [50] is applied in the following experiments. Given two classifiers A and B, GD is computed as follows:

$$GD(A; B|\mathcal{T}) = |win| - |loss|, \tag{11}$$

where $\mathcal{T}$ represents a set of datasets, $|win|$ and $|loss|$ are the number of datasets on which A performs better or worse than B, respectively. Figure 2 shows the fitting curve of GD between TAN$^m$ and RF in terms of zero-one loss. The X-axis denotes the index number of datasets, referred to as $\hat{r}$, which corresponds to that described in Table 2, and the Y-axis denotes the value of $GD(A; B|S_i)$, where $S_i$ is a set of datasets $\{D_1, \cdots, D_{\hat{i}} | \hat{i} < \hat{r}\}$. It can be seen from Figure 2 that there exists an obvious positive correlation between the values of $GD(TAN^m; RF|S_i)$ and the dataset size. While dealing with small datasets, such as `Lung-cancer`, `Labor` and `Wine`, TAN$^m$ enjoys a significant advantage over RF10, and the maximum value of GD reaches 7. A notable case is `Labor` dataset, where the zero-one loss result of TAN$^m$ is 0.0175 while that of RF10 is 0.0939. In general, we argue that the advantages for TAN$^m$ on small datasets can be attributed to that the model matching mechanism helps to fully mine the dependence relationships, especially those hidden in unlabeled test instances. With respect to the medium size datasets, TAN$^m$ also has a little advantage over RF100, and $GD(TAN^m; RF100|S_i)$ reaches a maximum of 9 when the data quantity is 4601 (`Sick` dataset). As referred to large datasets, RF100 provides competitive results with TAN$^m$. The 100 decision trees and more complex structures make RF100 obtain a better model fitting with the increase

**TABLE 2.** Detailed zero-one loss results of all BNCs.

| Dataset | NB | TAN | KDB | RF | LR | TAN$^m$ |
|---|---|---|---|---|---|---|
| Lung-cancer | 0.4375 | 0.5938 | 0.5938 | 0.5313 | 0.5578 | **0.4063** |
| Labor | 0.0351 | 0.0526 | **0.0175** | 0.0939 | 0.0737 | **0.0175** |
| Promoters | **0.0755** | 0.1321 | 0.1321 | 0.1792 | 0.1241 | 0.0943 |
| Lymphography | **0.1486** | 0.1757 | 0.1757 | 0.1824 | 0.3101 | 0.1622 |
| Iris | 0.0867 | 0.0800 | 0.0867 | **0.0400** | 0.0570 | 0.0733 |
| Teaching-ae | **0.4967** | 0.5497 | 0.5430 | 0.5270 | 0.5351 | 0.5166 |
| Wine | **0.0169** | 0.0337 | 0.0393 | 0.0225 | 0.0185 | **0.0169** |
| Sonar | 0.2308 | 0.2212 | 0.2308 | 0.2067 | **0.1784** | 0.2500 |
| Glass-id | 0.2617 | 0.2196 | 0.2243 | 0.2132 | **0.2007** | 0.2243 |
| New-thyroid | 0.0512 | 0.0651 | 0.0558 | **0.0372** | 0.0514 | 0.0698 |
| Audio | **0.2389** | 0.2920 | 0.3097 | 0.2803 | 0.2677 | 0.2788 |
| Heart | 0.1778 | 0.1926 | 0.1963 | 0.2037 | 0.1863 | **0.1741** |
| Hungarian | 0.1599 | 0.1701 | 0.1701 | 0.2007 | 0.1811 | **0.1395** |
| Soybean-large | 0.1238 | 0.1107 | **0.0912** | 0.1270 | 0.1530 | 0.1010 |
| Bupa | 0.4435 | 0.4435 | 0.4435 | **0.3159** | 0.3843 | 0.4435 |
| Dermatology | **0.0191** | 0.0328 | 0.0301 | 0.0410 | 0.0288 | 0.0219 |
| Horse-colic | 0.2174 | 0.2092 | 0.2174 | **0.1603** | 0.2726 | 0.2011 |
| House-votes-84 | 0.0943 | 0.0552 | 0.0690 | **0.0391** | 0.0493 | 0.0552 |
| Cylinder-bands | **0.2148** | 0.2833 | 0.2278 | 0.2667 | 0.2400 | 0.2519 |
| Chess | 0.1125 | 0.0926 | 0.0998 | 0.1071 | 0.1233 | **0.0871** |
| Credit-a | **0.1406** | 0.1507 | 0.1551 | 0.1493 | 0.1478 | 0.1449 |
| Crx | 0.1377 | 0.1478 | 0.1464 | 0.1581 | 0.1564 | **0.1333** |
| Breast-cancer-w | **0.0258** | 0.0415 | 0.0486 | 0.0415 | 0.0471 | 0.0358 |
| Vehicle | 0.3924 | 0.2943 | 0.3014 | **0.2530** | 0.2845 | 0.2849 |
| German | **0.2530** | 0.2730 | 0.2760 | 0.2684 | 0.2575 | 0.2620 |
| Segment | 0.0788 | 0.0390 | 0.0403 | 0.0413 | 0.0550 | **0.0355** |
| Hypothyroid | 0.0149 | 0.0104 | 0.0107 | **0.0094** | 0.0180 | 0.0095 |
| Splice-c4.5 | 0.0444 | 0.0466 | 0.0482 | 0.0489 | 0.0692 | **0.0409** |
| Kr-vs-kp | 0.1214 | 0.0776 | 0.0544 | **0.0128** | 0.0277 | 0.0735 |
| Dis | 0.0159 | 0.0159 | 0.0146 | 0.0132 | 0.0170 | **0.0130** |
| Hypo | 0.0138 | 0.0141 | 0.0077 | 0.0122 | **0.0062** | 0.0082 |
| Sick | 0.0308 | 0.0257 | **0.0241** | 0.0263 | 0.0269 | 0.0247 |
| Spambase | 0.1015 | 0.0669 | 0.0765 | **0.0575** | 0.0626 | 0.0669 |
| Waveform-5000 | 0.2006 | 0.1844 | 0.1820 | 0.1558 | **0.1470** | 0.1622 |
| Phoneme | 0.2615 | 0.2733 | 0.2120 | **0.1789** | 0.2544 | 0.2150 |
| Wall-following | 0.1054 | 0.0554 | 0.0462 | 0.0216 | **0.0118** | 0.0493 |
| Page-blocks | 0.0619 | 0.0415 | 0.0433 | **0.0309** | 0.0368 | 0.0354 |
| Optdigits | 0.0767 | 0.0407 | 0.0416 | 0.0458 | 0.0533 | **0.0390** |
| Thyroid | 0.1111 | 0.0720 | 0.0693 | 0.0750 | 0.0683 | **0.0678** |
| Pendigits | 0.1181 | 0.0321 | 0.0362 | 0.0339 | 0.0413 | **0.0254** |
| Magic | 0.2239 | 0.1675 | 0.1742 | 0.1674 | **0.1538** | 0.1733 |
| Letter-recog | 0.2525 | 0.1300 | 0.1286 | **0.0902** | 0.1639 | 0.1111 |
| Adult | 0.1592 | 0.1380 | 0.1385 | 0.1466 | **0.1274** | 0.1345 |
| Shuttle | 0.0039 | 0.0015 | 0.0015 | 0.0009 | **0.0004** | 0.0011 |
| Localization | 0.4955 | 0.3575 | 0.3706 | **0.2976** | 0.4584 | 0.3481 |
| Poker-hand | 0.4988 | 0.3295 | 0.3291 | N/A | 0.4988 | **0.0806** |

**TABLE 3.** Win/draw/loss comparison results between TAN$^m$ and RF in terms of zero-one loss.

| W/D/L | RF |
|---|---|
| TAN$^m$ | 21/9/15 |

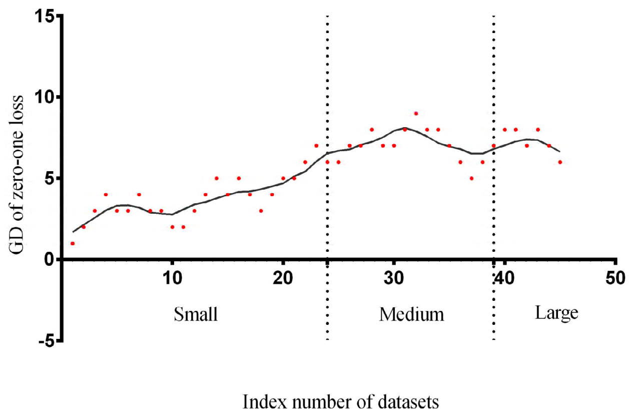


**FIGURE 2.** **The fitting curves of GD between TAN$^m$ and RF in terms of zero-one loss.**

**TABLE 4.** **Win/draw/loss comparison results between TAN$^m$ and LR in terms of zero-one loss.**

| W/D/L | LR |
|---|---|
| TAN$^m$ | 24/8/14 |

of data. However, the computational cost in terms of memory and time is much higher than that of TAN$^m$.

### 2) LOGISTIC REGRESSION

In this section, we compare the classification performance of TAN$^m$ with popular discriminative classifier Logistic regression (LR). LR in [47] is implemented for our experiments. The detailed results for each dataset in terms of zero-one loss can be found in Table 2. Table 4 compares the zero-one loss results of TAN$^m$ with respect to LR. It is encouraging to see that TAN$^m$ has lower zero-one loss than LR winning on 24, drawing on 8 and losing only on 14 datasets.

To make the experimental results more intuitive, Figure 3 presents the scatter plot of zero-one loss, where the X-axis and Y-axis represent the zero-one loss results of LR and TAN$^m$, respectively. As can be seen, there exist numerous scatters under the diagonal line, such as `Lymphography` dataset and `Poker-hand` dataset, which means that TAN$^m$ achieves significant advantages than LR on those datasets. A notable case is `Poker-hand` dataset where TAN$^m$ achieves a 83.8% zero-one loss reduction comparing with LR. Moreover, there are only three points that are clearly above the diagonal line, that is `Sonar`, `Bupa` and `Kr-vs-kp` dataset, which means that LR has a clear advantage over TAN$^m$ on these three datasets. That is to say, for the rest 11 of 14 datasets where TAN$^m$ losses, the classification accuracy of TAN$^m$ is close to that of LR. The experimental results show that, in fact, TAN$^m$ can beat LR on most datasets. As a classic discriminative algorithm, LR needs mass data to model the conditional probability $P(c|\mathbf{x})$ and learn the model parameters through maximising the conditional likelihood [51]. Although maximum conditional likelihood has desirable asymptotic properties, it may often lead to overfitting on training data, which may result in failure of fitting additional data or predicting unlabeled test instances

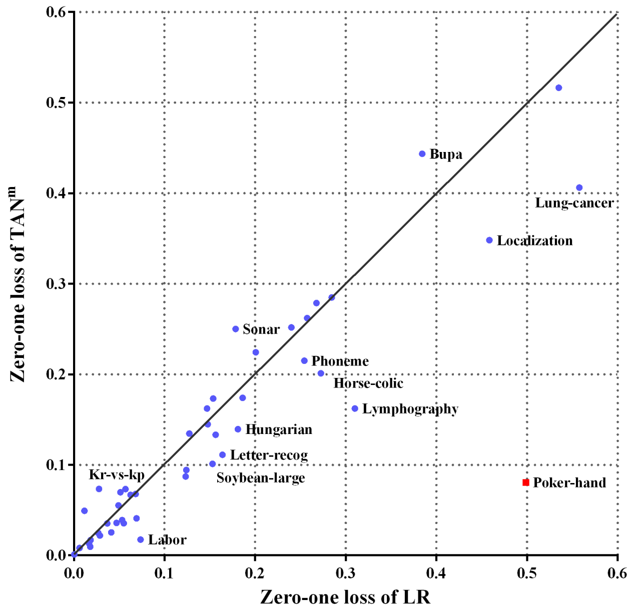


**FIGURE 3.** **The scatter plot when comparing the zero-one loss results between TAN$^m$ and LR.**

**TABLE 5.** **Win/draw/loss comparison results of zero-one loss on all datasets.**

| | NB | TAN | KDB |
|---|---|---|---|
| TAN | 23/7/16 | — | — |
| KDB | 26/4/16 | 11/27/8 | — |
| TAN$^m$ | 31/7/8 | 30/14/2 | 30/9/7 |

reliably [52], [53]. In contrast, by learning a set of submodels for each unlabeled test instance, TAN$^m$ can better mine the significant dependence relationships implicated in test instances, which may help to alleviate the negative effect caused by overfitting.

## B. TAN$^M$ VS CLASSIC BNCS

This set of experiments compare TAN$^m$ with three classic BNCs, that is NB, TAN, and KDB ($k = 2$). Tables 5 presents W/D/L records summarizing the zero-one loss results of the different approaches, according to the detailed results in Table 2. The results indicate that NB performs the worst among these BNCs. As the dependence degree or structure complexity increases, KDB achieves lower error than TAN on 11 datasets. It can be seen that TAN$^m$ has significantly better zero-one loss than all other BNCs. For example, TAN$^m$ achieves superior performance to NB, which shows 31 wins and 8 losses. When compared with KDB, TAN$^m$ achieves significant advantages and results in 30 wins. Most of all, TAN$^m$ achieves lower zero-one loss results more often than TAN (30 wins and only 2 losses).

For further analysis, given classifiers $A$ and $B$, the relative zero-one loss ratio (RZR) [54] is applied to compare the performance and defined as follows,

$$RZR(A/B) = 1 - \frac{\xi(A)}{\xi(B)}, \tag{12}$$

where $\xi(\cdot)$ represents the zero-one loss results. Obviously, a higher value of RZR(A/B) indicates the better performance of classifier $A$. Figure 4 compares TAN$^m$ with NB, TAN and KDB in terms of RZR. Each figure is divided into three parts by comparing dataset size. In diverse parts, different symbols and colors are applied to show different situations. From Figure 4(a), we can find that NB is competitive compared to TAN$^m$, when handling with small datasets. The reason lies in that the precise estimation of CMI is determined by probability estimation, which is affected greatly by dataset size and the robustness of network structure will be affected negatively by imprecise probability estimation. For instance, for `Promoters` dataset with 106 instances and 57 predictive attributes, it is almost impossible to ensure that the basic causal relationships learned are of a high confidence level, which causes that a simple structure can beat a complicated one. When compared with TAN, it can be seen from Figure 4(b) that there are only four points under the X-axis, which illustrates that the model matching mechanism did not reduce the classification error on these four datasets. However, the minimum value of RZR(TAN$^m$/TAN) is $-0.1302$ on `Sonar` dataset, which means that there is only a very small gap between TAN$^m$ and TAN on the four datasets where TAN outperforms TAN$^m$. Surprisingly, it has been shown from Figure 4(c) that the prediction accuracy of TAN$^m$ compares very well with KDB, especially when the data quantity is large, although KDB is a 2-dependence BNC. A notable case is our largest dataset `Porker-hand` where the value of RZR(TAN$^m$/KDB) reaches 0.7551.

### 1) BIAS AND VARIANCE

Kohavi and Wolpert [55] presented a bias-variance decomposition of zero-one loss from sampling theory statistics for analyzing different learning scenarios. In this part we show a set of experimental comparison of TAN$^m$ with the three BNCs in terms of bias and variance. Table 6 and 7 present the detailed results of bias and variance on all datasets. The corresponding W/D/L comparison results are shown in Table 8. We can observe that in terms of bias TAN$^m$ performs better than NB (26/9/11) and TAN (20/16/10). Due to its higher degree of dependence, KDB performs the best among all the BNCs. However, the advantages of KDB are not significant when compared with TAN$^m$ (19/12/15). In terms of variance, without a doubt, NB performs the best among all BNCs due to its definite network structure regardless of the change of training data. TAN$^m$ beats KDB in 37 datasets and losses in 7. This superiority is more obvious when comparing TAN$^m$ with TAN (38 wins and 2 losses). Thus the advantage of TAN$^m$ over other three BNCs in terms of zero-one loss can be attributed to the change in variance. The variance increases as the algorithm becomes more sensitive to the change in labeled training data. Obviously, the model matching mechanism helps to alleviate the negative effect caused by overfitting.

### 2) MATTHEWS CORRELATION COEFFICIENT

Since the model matching framework is proposed to alleviate the negative effect of the prior probability, one can expect

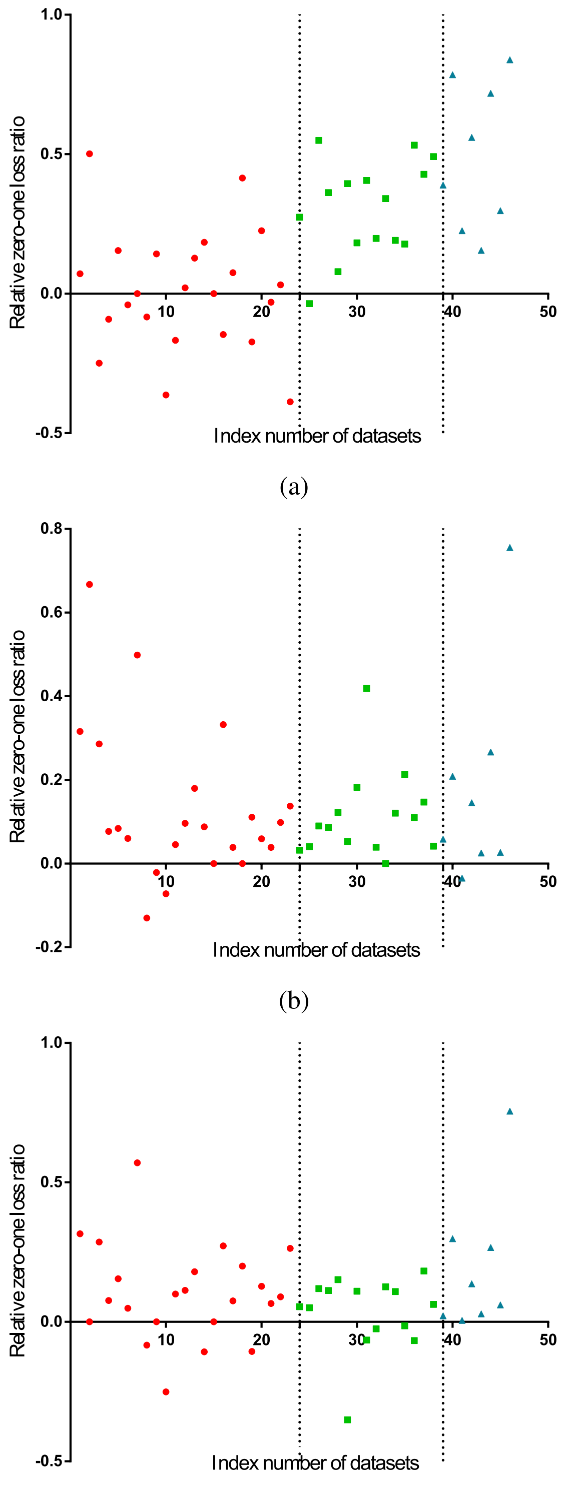


**FIGURE 4. The experimental results of relative zero-one loss ratio. (a) vs. NB. (b) vs. TAN. (c) vs. KDB.**

that TAN$^m$ achieves better performance on skewed datasets, where the class distribution is highly unbalanced. To verify this point, the Matthews correlation coefficient (MCC), which provides a balanced measure for skewed datasets by

**TABLE 6. Experimental results of average bias.**

| Dataset | NB | TAN | KDB | TAN$^m$ |
|---|---|---|---|---|
| Lung-cancer | **0.3330** | 0.3450 | 0.3490 | 0.3380 |
| Labor | 0.0289 | 0.0211 | 0.0279 | **0.0163** |
| Promoters | **0.0786** | 0.1329 | 0.1569 | 0.1031 |
| Lymphography | **0.0902** | 0.1027 | 0.1041 | 0.0973 |
| Iris | 0.0612 | 0.0638 | **0.0596** | 0.0706 |
| Teaching-ae | 0.4836 | 0.4566 | 0.4606 | **0.4478** |
| Wine | **0.0331** | 0.0507 | 0.0520 | 0.0436 |
| Sonar | 0.1672 | **0.1646** | 0.1686 | 0.1726 |
| Glass-id | 0.2901 | 0.2756 | **0.2713** | 0.2799 |
| New-thyroid | 0.0290 | **0.0277** | 0.0348 | 0.0296 |
| Audio | **0.2733** | 0.3617 | 0.3493 | 0.3176 |
| Heart | **0.1368** | 0.1472 | 0.1697 | 0.1397 |
| Hungarian | 0.1646 | 0.1424 | 0.1480 | **0.1421** |
| Soybean-large | **0.1070** | 0.1422 | 0.1086 | 0.1393 |
| Bupa | 0.2558 | 0.2558 | 0.2558 | 0.2558 |
| Dermatology | **0.0079** | 0.0274 | 0.0489 | 0.0273 |
| Horse-colic | 0.1966 | 0.1848 | **0.1689** | 0.1805 |
| House-votes-84 | 0.0899 | 0.0410 | **0.0258** | 0.0560 |
| Cylinder-bands | 0.2000 | 0.3117 | **0.1939** | 0.3388 |
| Chess | 0.1413 | 0.1437 | **0.1119** | 0.1256 |
| Credit-a | **0.0912** | 0.1171 | 0.1137 | 0.1124 |
| Crx | **0.0922** | 0.1180 | 0.1197 | 0.1090 |
| Breast-cancer-w | **0.0187** | 0.0384 | 0.0449 | 0.0195 |
| Vehicle | 0.3330 | **0.2382** | 0.2494 | 0.2549 |
| German | 0.2025 | 0.2057 | 0.2108 | **0.1989** |
| Segment | 0.0785 | 0.0491 | 0.0453 | **0.0444** |
| Hypothyroid | 0.0116 | 0.0104 | **0.0096** | 0.0099 |
| Splice-c4.5 | 0.0351 | 0.0395 | 0.0961 | **0.0324** |
| Kr-vs-kp | 0.1107 | 0.0702 | **0.0417** | 0.0642 |
| Dis | **0.0165** | 0.0193 | 0.0191 | 0.0194 |
| Hypo | 0.0092 | 0.0124 | 0.0077 | **0.0060** |
| Sick | 0.0246 | 0.0207 | **0.0198** | 0.0234 |
| Spambase | 0.0929 | 0.0570 | **0.0497** | 0.0621 |
| Waveform-5000 | 0.1762 | 0.1232 | **0.1157** | 0.1221 |
| Phoneme | 0.2216 | 0.2394 | **0.1572** | 0.2035 |
| Wall-following | 0.0951 | 0.0491 | **0.0257** | 0.0453 |
| Page-blocks | 0.0451 | 0.0308 | 0.0280 | **0.0279** |
| Optdigits | 0.0685 | 0.0275 | **0.0250** | 0.0365 |
| Thyroid | 0.0994 | 0.0587 | **0.0553** | 0.0627 |
| Pendigits | 0.1095 | 0.0314 | **0.0207** | 0.0285 |
| Magic | 0.2111 | 0.1252 | **0.1241** | 0.1334 |
| Letter-recog | 0.2207 | 0.1032 | **0.0806** | 0.1072 |
| Adult | 0.1649 | 0.1312 | **0.1220** | 0.1290 |
| Shuttle | 0.0040 | 0.0008 | 0.0007 | **0.0006** |
| Localization | 0.4523 | 0.3106 | **0.2134** | 0.2934 |
| Poker-hand | 0.4979 | 0.2865 | 0.1326 | **0.0770** |

**TABLE 7. Experimental results of average variance.**

| Dataset | NB | TAN | KDB | TAN$^m$ |
|---|---|---|---|---|
| Lung-cancer | **0.1970** | 0.2950 | 0.2810 | 0.2320 |
| Labor | 0.0395 | 0.0632 | 0.0721 | **0.0311** |
| Promoters | **0.0786** | 0.1729 | 0.1889 | 0.1226 |
| Lymphography | **0.0343** | 0.1116 | 0.1408 | 0.0598 |
| Iris | 0.0428 | 0.0662 | **0.0404** | 0.0494 |
| Teaching-ae | **0.1484** | 0.1914 | 0.1494 | 0.1902 |
| Wine | **0.0093** | 0.0493 | 0.0649 | 0.0225 |
| Sonar | **0.0907** | 0.1165 | 0.1199 | 0.0984 |
| Glass-id | **0.0930** | 0.1075 | 0.1189 | 0.1004 |
| New-thyroid | **0.0161** | 0.0272 | 0.0385 | 0.0254 |
| Audio | 0.1000 | **0.0983** | 0.1373 | 0.1024 |
| Heart | **0.0443** | 0.0739 | 0.0914 | 0.0614 |
| Hungarian | **0.0201** | 0.0596 | 0.0561 | 0.0426 |
| Soybean-large | **0.0783** | 0.1176 | 0.0982 | 0.0930 |
| Bupa | 0.1929 | 0.1929 | 0.1929 | 0.1929 |
| Dermatology | **0.0216** | 0.0513 | 0.0684 | 0.0325 |
| Horse-colic | **0.0353** | 0.1021 | 0.1384 | 0.0720 |
| House-votes-84 | **0.0066** | 0.0170 | 0.0197 | 0.0088 |
| Cylinder-bands | 0.0656 | 0.0739 | 0.0750 | **0.0534** |
| Chess | **0.0401** | 0.0486 | 0.0531 | 0.0460 |
| Credit-a | **0.0249** | 0.0555 | 0.0768 | 0.0484 |
| Crx | **0.0200** | 0.0520 | 0.0663 | 0.0471 |
| Breast-cancer-w | **0.0010** | 0.0337 | 0.0504 | 0.0195 |
| Vehicle | **0.1120** | 0.1299 | 0.1283 | 0.1320 |
| German | **0.0678** | 0.1009 | 0.1192 | 0.0902 |
| Segment | 0.0259 | 0.0294 | 0.0381 | **0.0218** |
| Hypothyroid | 0.0031 | 0.0034 | **0.0024** | 0.0032 |
| Splice-c4.5 | **0.0078** | 0.0289 | 0.0800 | 0.0172 |
| Kr-vs-kp | 0.0186 | 0.0152 | **0.0111** | 0.0212 |
| Dis | 0.0069 | 0.0005 | 0.0011 | **0.0004** |
| Hypo | **0.0051** | 0.0071 | 0.0069 | 0.0062 |
| Sick | 0.0047 | 0.0051 | **0.0043** | 0.0051 |
| Spambase | **0.0092** | 0.0158 | 0.0214 | 0.0122 |
| Waveform-5000 | **0.0259** | 0.0690 | 0.0843 | 0.0440 |
| Phoneme | 0.1215 | 0.1828 | **0.1064** | 0.1399 |
| Wall-following | **0.0211** | 0.0288 | 0.0294 | 0.0230 |
| Page-blocks | 0.0135 | 0.0143 | 0.0177 | **0.0122** |
| Optdigits | 0.0153 | 0.0185 | 0.0254 | **0.0126** |
| Thyroid | **0.0205** | 0.0257 | 0.0272 | 0.0221 |
| Pendigits | 0.0157 | 0.0200 | 0.0236 | **0.0111** |
| Magic | **0.0174** | 0.0490 | 0.0491 | 0.0399 |
| Letter-recog | 0.0471 | 0.0591 | 0.0709 | **0.0417** |
| Adult | **0.0069** | 0.0165 | 0.0285 | 0.0145 |
| Shuttle | 0.0009 | 0.0004 | **0.0003** | 0.0004 |
| Localization | **0.0460** | 0.0594 | 0.1099 | 0.0672 |
| Poker-hand | **0.0000** | 0.0424 | 0.0633 | 0.0038 |

**TABLE 8. Win/Draw/Loss comparison results of bias and variance on all datasets.**

| | BNC | NB | TAN | KDB |
|---|---|---|---|---|
| Bias | TAN | 23/8/5 | - | - |
| | KDB | 25/9/12 | 23/16/7 | - |
| | TAN$^m$ | 26/9/11 | 20/16/10 | 15/12/19 |
| Variance | TAN | 3/2/41 | - | - |
| | KDB | 7/2/37 | 9/8/29 | - |
| | TAN$^m$ | 9/3/34 | 38/6/2 | 37/2/7 |

taking into account the class distribution [56], is applied in this part of experiments. Since there are many multi-class datasets in our experiments, we regard MCC as the extended MCC [57] in the following discussion. The classification results can be shown in the form of a confusion matrix as follows:

$$\begin{bmatrix} N_{11} & \cdots & N_{1m} \\ \vdots & \ddots & \vdots \\ N_{m1} & \cdots & N_{mm} \end{bmatrix} \tag{13}$$

Each entry $N_{ii}$ of the matrix gives the number of instances, whose true class was $C_i$, that were actually assigned to $C_i$, where $1 \leq i \leq m$. Each entry $N_{ij}$ of the matrix gives the number of instances, whose true class was $C_i$, that were actually assigned to $C_j$, where $i \neq j$ and $1 \leq i, j \leq m$. Given the confusion matrix, the extended MCC can be calculated as follow,

$$MCC = \frac{\sum_{mij} N_{ii}N_{jm} - N_{ij}N_{mi}}{\sqrt{\sum_i(\sum_j N_{ij})(\sum_{j',i'\neq i} N_{i'j'})}\sqrt{\sum_i(\sum_j N_{ji})(\sum_{j',i'\neq i} N_{j'i'})}} \tag{14}$$

Note, the MCC reaches its best value at 1 which represents a perfect prediction and worst value at $-1$ which indicates a

**TABLE 9. Experimental results of average MCC.**

| Dataset | NB | TAN | KDB | TAN$^m$ |
|---|---|---|---|---|
| Lung-cancer | 0.3290 | 0.1012 | 0.0922 | **0.3806** |
| Labor* | 0.9273 | 0.8864 | **0.9626** | 0.9618 |
| Promoters | **0.8515** | 0.7380 | 0.7364 | 0.8119 |
| Lymphography | **0.7202** | 0.6633 | 0.6635 | 0.6893 |
| Iris | 0.8701 | 0.8800 | 0.8701 | **0.8901** |
| Teaching-ae | **0.2552** | 0.1774 | 0.1887 | 0.2261 |
| Wine | 0.9745 | 0.9488 | 0.9404 | **0.9748** |
| Sonar* | 0.5355 | **0.5557** | 0.5371 | 0.4977 |
| Glass-id | 0.5986 | **0.6631** | 0.6565 | 0.6558 |
| New-thyroid | **0.8889** | 0.8593 | 0.8782 | 0.8476 |
| Audio | **0.7205** | 0.6561 | 0.6355 | 0.6718 |
| Heart* | 0.6400 | 0.6085 | 0.6009 | **0.6463** |
| Hungarian* | 0.6475 | 0.6252 | 0.6272 | **0.6934** |
| Soybean-large | 0.8677 | 0.8797 | **0.9007** | 0.8913 |
| Bupa* | 0.0166 | 0.0166 | 0.0166 | 0.0166 |
| Dermatology | **0.9763** | 0.9590 | 0.9624 | 0.9728 |
| Horse-colic* | 0.5414 | 0.5466 | 0.5295 | **0.5672** |
| House-votes-84* | 0.8065 | 0.8839 | 0.8553 | **0.8848** |
| Cylinder-bands* | **0.5686** | 0.4180 | 0.5293 | 0.4933 |
| Chess* | 0.6922 | 0.7482 | 0.7280 | **0.7632** |
| Credit-a* | **0.7149** | 0.6942 | 0.6853 | 0.7061 |
| Crx* | 0.7215 | 0.7002 | 0.7036 | **0.7301** |
| Breast-cancer-w* | **0.9447** | 0.9078 | 0.8919 | 0.9216 |
| Vehicle | 0.4860 | 0.6094 | 0.5991 | **0.6245** |
| German* | **0.3783** | 0.3312 | 0.3077 | 0.3423 |
| Segment | 0.9082 | 0.9546 | 0.9531 | **0.9586** |
| Hypothyroid* | 0.8539 | 0.8849 | 0.8818 | **0.8978** |
| Splice-c4.5 | 0.9279 | 0.9241 | 0.9215 | **0.9335** |
| Kr-vs-kp* | 0.7567 | 0.8453 | **0.8910** | 0.8532 |
| Dis* | **0.4922** | 0.1842 | 0.3002 | 0.4250 |
| Hypo* | 0.9054 | 0.9007 | **0.9467** | 0.9434 |
| Sick* | 0.7541 | 0.7750 | **0.7950** | 0.7896 |
| Spambase* | 0.7862 | **0.8593** | 0.8393 | **0.8593** |
| Waveform-5000 | 0.7189 | 0.7234 | 0.7273 | **0.7587** |
| Phoneme | 0.7207 | 0.7084 | **0.7734** | 0.7703 |
| Wall-following | 0.8454 | 0.9170 | **0.9308** | 0.9265 |
| Page-blocks* | 0.7351 | 0.8062 | 0.8004 | **0.8294** |
| Optdigits | 0.9149 | 0.9547 | 0.9538 | **0.9568** |
| Thyroid | 0.7724 | 0.8378 | 0.8437 | **0.8503** |
| Pendigits | 0.8699 | 0.9643 | 0.9598 | **0.9719** |
| Magic* | 0.4894 | **0.6234** | 0.6080 | 0.6109 |
| Letter-recog | 0.7378 | 0.8648 | 0.8664 | **0.8846** |
| Adult* | 0.6069 | 0.6132 | 0.6112 | **0.6204** |
| Shuttle* | 0.9891 | 0.9957 | 0.9957 | **0.9968** |
| Localization | 0.3660 | 0.5474 | 0.5318 | **0.5580** |
| Poker-hand | 0.0000 | 0.4242 | 0.4248 | **0.8575** |

total disagreement between the predicted and observed classifications. Table 9 shows the average results of MCC on all datasets and the 23 datasets with data skewness are annotated with the symbol "*". The corresponding W/D/L comparison results are presented in Table 10. With respect to balanced datasets, we can easily find that TAN$^m$ always performs best. TAN$^m$ performs much better than TAN (5/18/0) and KDB (5/18/0). This superiority shows more obvious when comparing TAN$^m$ with NB (9 wins). For skewed datasets, as expected, we can see that TAN$^m$ still has the best MCC results among all the BNCs. For instance, when compared with NB, the TAN$^m$ wins on 7 datasets. TAN$^m$ also provides higher MCC results on the skewed datasets than TAN (5 wins and 1 loss) and KDB (5 wins and 2 losses). Hence, we can conduct that the model learning framework helps TAN$^m$ to have capacity to cope better with skewed datasets than other BNCs.

**TABLE 10. Win/Draw/Loss comparison results of MCC on all datasets.**

| | BNC | NB | TAN | KDB |
|---|---|---|---|---|
| Skewed datasets | TAN | 6/14/3 | - | - |
| | KDB | 6/12/5 | 4/18/1 | - |
| | TAN$^m$ | 7/12/4 | 5/17/1 | 5/16/2 |
| Balanced datasets | TAN | 8/10/5 | - | - |
| | KDB | 9/9/5 | 2/20/1 | - |
| | TAN$^m$ | 12/9/2 | 5/18/0 | 5/18/0 |

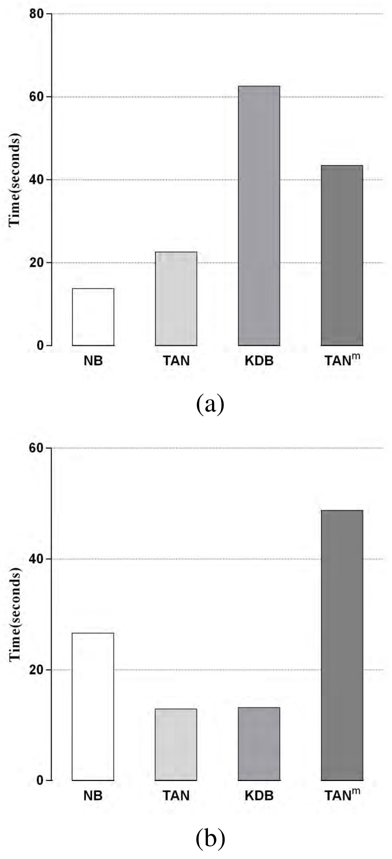


**FIGURE 5.** Time comparisons of NB, TAN, KDB and TAN$^m$. (a) Training times. (b) Classification times.

### 3) TIME COMPARISONS

In this part, we compare TAN$^m$ with other classic BNCs in terms of time consumption. Figure 5 shows training and classification time comparisons of NB, TAN, KDB and TAN$^m$. All experiments are conducted on a desktop computer with an Intel(R) Core(TM) i3-6100 CPU @ 3.70 GHz, 64 bits and 4096 MB of memory. Each bar represents the sum of time on 46 datasets in a 10-fold cross-validation experiment. No parallelization techniques have been used in any case. Figure 5(a) indicates that TAN$^m$ requires a bit more time for training than NB and TAN, since a set of TAN$^{\mathcal{T}}$ must be built in the training phase. Moreover, it is obvious that TAN$^m$ enjoys a significant advantage over KDB in terms of training time. With respect to classification time, TAN$^m$ takes a little more time than the other three BNCs. The reason lies in that TAN$^m$ learns a set of submodels for each unlabeled test instance, while other BNCs only need to directly calculate the joint probabilities. In general, model matching framework helps to significantly improve the classification performance of its base classifier at the cost of a small increase in time consumption, which is perfectly acceptable.

### *C. GLOBAL COMPARISON OF ALL LEARNERS*

In this section we analyze how all the learners explored in this paper perform when they are compared as a set. First, we apply the Friedman test for comparison of all learners. Friedman test [58] is a non-parametric measure, which is used to assess the whole classification performance of diverse algorithms. It ranks the algorithms for each dataset separately: algorithm that has the best performance getting the rank of 1, the second best rank 2, $\cdots$. In case of ties, average ranks are assigned. Table 11 shows the detailed results of rank on all datasets. Note that, although we can not get the results of RF on the `Poker-hand` dataset, it still ranks 1 due to its superior performance on the large datasets. The Friedman statistic can be calculated as follows,

$$F_F = \frac{(D-1)\chi_F^2}{D(g-1)-\chi_F^2}, \tag{15}$$

where

$$\chi_F^2 = \frac{12D}{g(g+1)}\left(\sum_i R_i^2 - \frac{g(g+1)^2}{4}\right), \tag{16}$$

$g$ is the number of algorithms being compared, $D$ is the number of datasets and $R_i$ is the average rank of the $i$-th algorithm. The null hypothesis of Friedman test is that there is no significant difference in average ranks. With 6 algorithms and 46 datasets, the Friedman test is distributed according to the $F$ distribution with $6-1=5$ and (6-1) $\times$ (46-1) $=$ 225 degrees of freedom. The critical value of $F$ (5,225) for $\alpha = 0.05$ is 2.2541. The result of Friedman test for zero-one loss, $F_F = 27.73 > 2.2541$ with $\rho < 0.001$. Hence, the null hypotheses is rejected. Figure 6 plots the average ranks across all datasets, along with the standard deviation for each learner. When assessing performance using zero-one loss, TAN$^m$ obtains the lowest average rank of 2.4348, followed by LR with 3.4674 and RF with 3.0978. The average ranks of KDB, TAN and NB are very close, that is 3.9457, 4.0000 and 4.0543, respectively. Surprisingly, we can find that the largest standard deviation is observed for NB, which usually ranks either very well or rather poorly among the datasets. From Figure 6, we can finally conduct that the advantage of TAN$^m$ is proved from the perspective of Friedman test.

To identify where exactly these differences are found, we run a set of posterior Nemenyi tests [59], which help to evaluate the significant difference between each pair of algorithms. Let $d_{mn}$ be the difference between the average ranks of the $m$-th algorithm and $n$-th algorithm. If $d_{mn} >$ critical difference (CD), we consider the difference between the algorithms is significant. The value of CD can be computed as follows,

$$CD = q_\alpha\sqrt{\frac{g(g+1)}{6D}}, \tag{17}$$

**TABLE 11.** Ranks of different algorithms on all datasets.

| Dataset | NB | TAN | KDB | RF | LR | TAN$^m$ |
|---|---|---|---|---|---|---|
| Lung-cancer | 2.0 | 5.5 | 5.5 | 3.0 | 4.0 | **1.0** |
| Labor | 3.0 | 4.0 | **1.5** | 6.0 | 5.0 | **1.5** |
| Promoters | **1.0** | 4.5 | 4.5 | 6.0 | 3.0 | 2.0 |
| Lymphography | **1.0** | 3.5 | 3.5 | 5.0 | 6.0 | 2.0 |
| Iris | 5.5 | 4.0 | 5.5 | **1.0** | 2.0 | 3.0 |
| Teaching-ae | **1.0** | 6.0 | 5.0 | 3.0 | 4.0 | 2.0 |
| Wine | **1.5** | 5.0 | 6.0 | 4.0 | 3.0 | **1.5** |
| Sonar | 4.5 | 3.0 | 4.5 | 2.0 | **1.0** | 6.0 |
| Glass-id | 6.0 | 3.0 | 4.5 | 2.0 | **1.0** | 4.5 |
| New-thyroid | 2.0 | 5.0 | 4.0 | **1.0** | 3.0 | 6.0 |
| Audio | **1.0** | 5.0 | 6.0 | 4.0 | 2.0 | 3.0 |
| Heart | 2.0 | 4.0 | 5.0 | 6.0 | 3.0 | **1.0** |
| Hungarian | 2.0 | 3.5 | 3.5 | 6.0 | 5.0 | **1.0** |
| Soybean-large | 4.0 | 3.0 | **1.0** | 5.0 | 6.0 | 2.0 |
| Bupa | 4.5 | 4.5 | 4.5 | **1.0** | 2.0 | 4.5 |
| Dermatology | **1.0** | 5.0 | 4.0 | 6.0 | 3.0 | 2.0 |
| Horse-colic | 4.5 | 3.0 | 4.5 | **1.0** | 6.0 | 2.0 |
| House-votes-84 | 6.0 | 3.5 | 5.0 | **1.0** | 2.0 | 3.5 |
| Cylinder-bands | **1.0** | 6.0 | 2.0 | 5.0 | 3.0 | 4.0 |
| Chess | 5.0 | 2.0 | 3.0 | 4.0 | 6.0 | **1.0** |
| Credit-a | **1.0** | 5.0 | 6.0 | 4.0 | 3.0 | 2.0 |
| Crx | 2.0 | 4.0 | 3.0 | 6.0 | 5.0 | **1.0** |
| Breast-cancer-w | **1.0** | 3.5 | 6.0 | 3.5 | 5.0 | 2.0 |
| Vehicle | 6.0 | 4.0 | 5.0 | **1.0** | 2.0 | 3.0 |
| German | **1.0** | 5.0 | 6.0 | 4.0 | 2.0 | 3.0 |
| Segment | 6.0 | 2.0 | 3.0 | 4.0 | 5.0 | **1.0** |
| Hypothyroid | 5.0 | 3.0 | 4.0 | **1.0** | 6.0 | 2.0 |
| Splice-c4.5 | 2.0 | 3.0 | 4.0 | 5.0 | 6.0 | **1.0** |
| Kr-vs-kp | 6.0 | 5.0 | 3.0 | **1.0** | 2.0 | 4.0 |
| Dis | 4.5 | 4.5 | 3.0 | 2.0 | 6.0 | **1.0** |
| Hypo | 5.0 | 6.0 | 2.0 | 4.0 | **1.0** | 3.0 |
| Sick | 6.0 | 3.0 | **1.0** | 4.0 | 5.0 | 2.0 |
| Spambase | 6.0 | 3.5 | 5.0 | **1.0** | 2.0 | 3.5 |
| Waveform-5000 | 6.0 | 5.0 | 4.0 | 2.0 | **1.0** | 3.0 |
| Phoneme | 5.0 | 6.0 | 2.0 | **1.0** | 4.0 | 3.0 |
| Wall-following | 6.0 | 5.0 | 3.0 | 2.0 | **1.0** | 4.0 |
| Page-blocks | 6.0 | 4.0 | 5.0 | **1.0** | 3.0 | 2.0 |
| Optdigits | 6.0 | 2.0 | 3.0 | 4.0 | 5.0 | **1.0** |
| Thyroid | 6.0 | 4.0 | 3.0 | 5.0 | 2.0 | **1.0** |
| Pendigits | 6.0 | 2.0 | 4.0 | 3.0 | 5.0 | **1.0** |
| Magic | 6.0 | 3.0 | 5.0 | 2.0 | **1.0** | 4.0 |
| Letter-recog | 6.0 | 4.0 | 3.0 | **1.0** | 5.0 | 2.0 |
| Adult | 6.0 | 3.0 | 4.0 | 5.0 | **1.0** | 2.0 |
| Shuttle | 6.0 | 4.5 | 4.5 | 2.0 | **1.0** | 3.0 |
| Localization | 6.0 | 3.0 | 4.0 | **1.0** | 5.0 | 2.0 |
| Poker-hand | 5.5 | 4.0 | 3.0 | **1.0** | 5.5 | 2.0 |
| AVG | 4.0543 | 4.0000 | 3.9457 | 3.0978 | 3.4674 | **2.4348** |

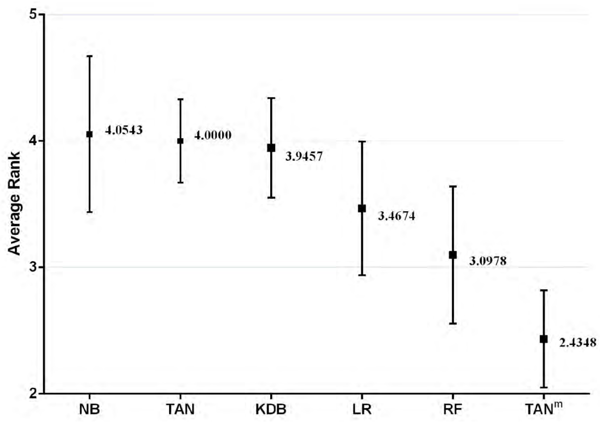


**FIGURE 6.** Average ranks in terms of zero-one loss for all algorithms.

where the critical value $q_\alpha$ for $\alpha = 0.05$ and $g = 6$ is 2.850 [60]. The CD for $\alpha = 0.05$ with 6 algorithms and 46 datasets is $CD = 2.850 \times \sqrt{6 \times (6+1)/(6 \times 46)} = 1.111$. The learners in Figure 7 are plotted on the red line

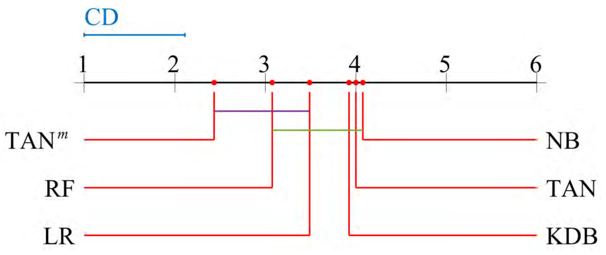


**FIGURE 7.** The results of Nemenyi test in terms of zero-one loss for all algorithms.

**TABLE 12.** Datasets.

| No. | Dataset | Instance | Attribute | Class |
|---|---|---|---|---|
| 1 | Labor | 57 | 16 | 2 |
| 2 | Promoters | 106 | 57 | 2 |
| 3 | Sonar | 208 | 61 | 2 |
| 4 | Heart | 270 | 13 | 2 |
| 5 | Hungarian | 294 | 14 | 2 |
| 6 | Bupa | 345 | 6 | 2 |
| 7 | Horse-colic | 368 | 21 | 2 |
| 8 | House-votes-84 | 435 | 17 | 2 |
| 9 | Cylinder-bands | 540 | 39 | 2 |
| 10 | Chess | 551 | 40 | 2 |
| 11 | Credit-a | 690 | 15 | 2 |
| 12 | Crx | 690 | 15 | 2 |
| 13 | Breast-cancer-w | 699 | 9 | 2 |
| 14 | German | 1000 | 20 | 2 |
| 15 | Hypothyroid | 3163 | 25 | 2 |
| 16 | Kr-vs-kp | 3196 | 36 | 2 |
| 17 | Dis | 3772 | 29 | 2 |
| 18 | Sick | 3772 | 29 | 2 |
| 19 | Spambase | 4601 | 57 | 2 |
| 20 | Magic | 19020 | 10 | 2 |
| 21 | Adult | 48842 | 14 | 2 |
| 22 | Lung-cancer | 32 | 56 | 3 |
| 23 | Iris | 150 | 4 | 3 |
| 24 | Teaching-ae | 151 | 5 | 3 |
| 25 | Wine | 178 | 13 | 3 |
| 26 | Glass-id | 214 | 9 | 3 |
| 27 | New-thyroid | 215 | 5 | 3 |
| 28 | Splice-c4.5 | 3177 | 60 | 3 |
| 29 | Waveform-5000 | 5000 | 41 | 3 |
| 30 | Lymphography | 148 | 18 | 4 |
| 31 | Vehicle | 846 | 18 | 4 |
| 32 | Hypo | 3772 | 29 | 4 |
| 33 | Wall-following | 5456 | 24 | 4 |
| 34 | Page-blocks | 5473 | 10 | 5 |
| 35 | Dermatology | 366 | 34 | 6 |
| 36 | Segment | 2310 | 19 | 7 |
| 37 | Shuttle | 58000 | 9 | 7 |
| 38 | Optdigits | 5620 | 64 | 10 |
| 39 | Pendigits | 10992 | 16 | 10 |
| 40 | Poker-hand | 1025010 | 10 | 10 |
| 41 | Localization | 164860 | 5 | 11 |
| 42 | Soybean-large | 307 | 35 | 19 |
| 43 | Thyroid | 9169 | 29 | 20 |
| 44 | Audio | 226 | 69 | 24 |
| 45 | Letter-recog | 20000 | 16 | 26 |
| 46 | Phoneme | 5438 | 8 | 52 |

on the basis of their average ranks, which are corresponding to the red nodes on the top black line. If two algorithms has no significant difference, they are connected by a line. From Figure 7, we can easily find that the average rank of TAN$^m$ is significantly lower than those of NB, TAN and KDB. TAN$^m$ also achieves lower average rank than RF and LR, but not significantly so.

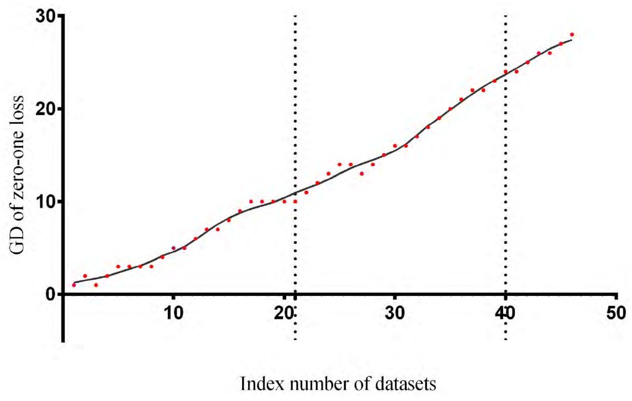


**FIGURE 8.** The fitting curve of GD between TAN$^m$ and TAN in terms of zero-one loss.

## D. DISCUSSION

To further explain the effectiveness of our proposed model matching framework, we compare the zero-one loss results of TAN$^m$ and its base classifier (TAN) on datasets with various $z$ class labels. Figure 8 shows the fitting curve of GD between TAN$^m$ and TAN in terms of zero-one loss. Note that the X-axis represents the index number of datasets ordered by $z$ and dataset size. The details of each dataset are shown in Table 12. Two dotted lines divide the figure into three parts, each part is associated to datasets with $z = 2$, $3 \leq z < 10$ and $z \geq 10$. When $z = 2$, the advantage of TAN$^m$ over TAN is not obvious. That is, TAN$^m$ draws on 9 out of 21 datasets compared with TAN. With respect to datasets with $3 \leq z < 10$, it is encouraging to see that TAN$^m$ shows a significant advantage over TAN, winning on 14 and drawing on 3 datasets. With the increase of $z$, on 7 datasets with $z \geq 10$, TAN$^m$ exhibits significantly higher accuracy than TAN on 6 datasets. It also can be seen from Figure 8 that TAN$^m$ never loses on any datasets when $z \geq 4$. Finally, the maximum value of $GD(TAN^m; TAN|S_i)$ reaches 28.

Especially, there are 4 datasets with more than 20 class labels in our experiments, that are `Thyroid`, `Audio`, `Letter-recog` and `Phoneme`. Most importantly, TAN$^m$ enjoys a significant advantage over TAN with 3 wins and only 1 draw. On `Phoneme` dataset with 52 class labels, it is worth mentioning that RZR(TAN$^m$/TAN) reaches 0.2133, which is much higher than the average of RZR, i.e. 0.1333. To conclude, the model matching framework helps BNCs have capacity to cope better with multiclass problem than its base classifiers.

## V. CONCLUSIONS

In machine learning, most research has focused primarily on dealing with labeled data with classifiers and unlabeled ones with clusterings. In this paper, we propose a novel framework, called model matching, that attempts to use "clustering" strategy to handle the classification problem. A set of BNC$^{\mathcal{T}}$s is built for different clusters in each of which the instances are assigned with the same class label and the corresponding set of BNC$^p$s is learned for each unlabeled test instance. The cross entropy method is applied to compare the structure similarity of BNC$^{\mathcal{T}}$ and BNC$^p$ to make the final prediction. Extensive experimental results show that model matching significantly improves the generalization performance of base classifiers. Exploration of application of model matching in other kinds of machine learning techniques, e.g., decision tree or support vector machine, is a further area for future work.

## REFERENCES

[1] Y. Yan, Q. Wu, M. Tan, M. K. Ng, H. Min, and I. W. Tsang, "Online heterogeneous transfer by hedge ensemble of offline and online decisions," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3252–3263, Jul. 2018.

[2] Q. Wu, H. Wu, X. Zhou, M. Tan, Y. Xu, Y. Yan, and T. Hao, "Online transfer learning with multiple homogeneous or heterogeneous sources," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 7, pp. 1494–1507, Jul. 2017.

[3] W. Qingyao, T. Mingkui, S. Hengjie, C. Jian, and M. K. Ng, "ML-FOREST: A multi-label tree ensemble method for multi-label classification," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 10, pp. 2665–2680, Oct. 2016.

[4] Q. Wu, Y. Ye, H. Zhang, T. W. S. Chow, and S.-S. Ho, "ML-TREE: A tree-structure-based approach to multilabel learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 430–443, Mar. 2015.

[5] C. L. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Inf. Sci.*, vol. 275, pp. 314–347, Aug. 2014.

[6] K. Nguyen, T. Le, T. D. Nguyen, D. Phung, and G. I. Webb, "Robust Bayesian kernel machine via stein variational gradient descent for big data," in *Proc. 24th SIGKDD*, London, U.K., 2018, pp. 2003–2011.

[7] T. Vandal, E. Kodra, J. Dy, S. Ganguly, R. Nemani, and A. R. Ganguly, "Quantifying uncertainty in discrete-continuous and skewed data with Bayesian deep learning," in *Proc. 24th SIGKDD*, London, U.K., 2018, pp. 2377–2386.

[8] L. Zhou, L. Wang, L. Liu, P. Ogunbona, and D. Shen, "Learning discriminative Bayesian networks from high-dimensional continuous neuroimaging data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2269–2283, Jun. 2015.

[9] F. Pernkopf and M. Wohlmayr, "Stochastic margin-based structure learning of Bayesian network classifiers," *Pattern Recognit*, vol. 46, no. 2, pp. 464–471, Aug. 2013.

[10] F. Pernkopf, M. Wohlmayr, and S. Tschiatschek, "Maximum margin Bayesian network classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 521–532, Mar. 2012.

[11] M. Bartlett and J. Cussens, "Integer linear programming for the Bayesian network structure learning problem," *Artif. Intell.*, vol. 244, pp. 258–271, Mar. 2017.

[12] N. A. Zaidi, J. Cerquides, M. J. Carman, and G. I. Webb, "Alleviating naive Bayes attribute independence assumption by attribute weighting," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 1947–1988, Jul. 2013.

[13] G. Varando, C. Bielza, and P. Larrañaga, "Decision boundary for discrete Bayesian network classifiers," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 2725–2749, Dec. 2015.

[14] F. Zheng, G. I. Webb, P. Suraweera, and L. Zhu, "Subsumption resolution: An efficient and effective technique for semi-naive Bayesian learning," *Mach. Learn.*, vol. 87, no. 1, pp. 93–125, Dec. 2012.

[15] S.-C. Wang, R. Gao, and L.-M. Wang, "Bayesian network classifiers based on Gaussian kernel density," *Expert Syst. Appl.*, vol. 51, no. 1, pp. 207–217, Jun. 2016.

[16] A. M. Martinez, G. I. Webb, S. L. Chen, and N. A. Zaidi, "Scalable learning of Bayesian network classifiers," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1–35, Apr. 2013.

[17] L. Jiang, S. Wang, C. Li, and L. Zhang, "Structure extended multinomial naive Bayes," *Inf. Sci.*, vol. 329, pp. 346–356, Feb. 2016.

[18] C. Bielza and P. Larrañaga, "Discrete Bayesian network classifiers: A survey," *ACM Comput. Surv.*, vol. 47, no. 1, Jul. 2014, Art. no. 5.

[19] S. Chen, A. M. Martínez, G. I. Webb, and L. Wang, "Selective A*n* DE for large data learning: A low-bias memory constrained approach," *Knowl. Inf. Syst.*, vol. 50, pp. 475–503, Feb. 2017.

[20] S. Chen, A. M. Martínez, G. I. Webb, and L. Wang, "Sample-based attribute selective A*n* DE for large data," *IEEE Trans. Knowl. Data. Eng.*, vol. 29, no. 1, pp. 172–185, Jan. 2016.

[21] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Mach. Learn.*, vol. 29, no. 2, pp. 131–163, Nov. 1997.

[22] M. Sahami, "Learning limited dependence Bayesian classifiers," in *Proc. 2nd SIGKDD*, Portland, OR, USA, 1996, pp. 335–338.

[23] M. Hall, "A decision tree-based attribute weighting filter for naive Bayes," *Knowl.-Based Syst.*, vol. 20, no. 2, pp. 120–126, Mar. 2007.

[24] J. Wu and Z. Cai, "Attribute weighting via differential evolution algorithm for attribute weighted naive Bayes (WNB)," *J. Comput. Inf. Syst.*, vol. 7, no. 5, pp. 1672–1679, Mar. 2011.

[25] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ, USA: Wiley, 2012, pp. 1–22.

[26] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, Aug. 2015.

[27] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Inf. Sci.*, vols. 409–410, pp. 17–26, Oct. 2017.

[28] H. L. Nguyen, Y.-K. Woon, and W.-K. Ng, "A survey on data stream clustering and classification," *Knowl. Inf. Syst.*, vol. 45, no. 3, pp. 535–569, Dec. 2015.

[29] P. D. Mcnicholas, "Model-based clustering," *J. Classif*, vol. 33, pp. 1–43, Nov. 2016.

[30] T. Bdiri, N. Bouguila, and D. Ziou, "Variational Bayesian inference for infinite generalized inverted Dirichlet mixtures with feature selection and its application to clustering," *Appl. Intell.*, vol. 44, no. 3, pp. 507–525, Nov. 2016.

[31] R. P. Browne, P. D. McNicholas, and M. D. Sparling, "Model-based learning using a mixture of mixtures of Gaussian and uniform distributions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 814–817, Apr. 2012.

[32] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA, USA: Morgan Kaufmann, 1988.

[33] S. Jiang and H. Zhang, "Full Bayesian network classifiers," in *Proc. 23rd ICML*, Pittsburgh, PA, USA, 2006, pp. 897–904.

[34] G. I. Webb, J. R. Boughton, and Z. Wang, "Not so naive Bayes: Aggregating one-dependence estimators," *Mach. Learn.*, vol. 58, no. 1, pp. 5–24, Jan. 2005.

[35] Y. Zhao, Y. Chen, K. Tu, and J. Tian, "Learning Bayesian network structures under incremental construction curricula," *Neurocomputing*, vol. 258, pp. 30–40, Oct. 2017.

[36] L. Jiang, Z. Cai, D. Wang, and H. Zhang, "Improving tree augmented naive Bayes for class probability estimation," *Knowl.-Based Syst.*, vol. 26, pp. 239–245, Feb. 2012.

[37] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Mach. Learn.*, vol. 29, nos. 2–3, pp. 103–130, Nov. 1997.

[38] M. G. Madden, "On the classification performance of TAN and general Bayesian networks," *Knowl.-Based Syst.*, vol. 22, pp. 489–495, Jan. 2009.

[39] G. A. Ruz and D. T. Pham, "Building Bayesian network classifiers through a Bayesian complexity monitoring system," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 223, no. 3, pp. 743–755, Mar. 2009.

[40] M. Zaffalon and E. Fagiuoli, "Tree-based credal networks for classification," *Reliable Comput.*, vol. 9, no. 6, pp. 487–509, Mar. 2003.

[41] S.-C. Ma and H.-B. Shi, "Tree-augmented naive Bayes ensembles," in *Proc. 3rd ICML*, Shanghai, China, Aug. 2004, pp. 1497–1502.

[42] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[43] S.-H. Park and J. Fürnkranz, "Efficient implementation of class-based decomposition schemes for Naïve Bayes," *Mach. Learn.*, vol. 96, no. 3, pp. 295–309, Sep. 2014.

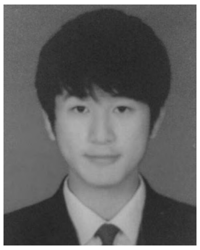
[44] C. Farabet, C. Couprie, M. Najman, and Y. LeCun, "Scene parsing with multiscale feature learning, purity trees, and optimal covers," in *Proc. ICML*, 2012, pp. 575–582.

[45] P. M. Murphy and D. W. Aha. *UCI Repository of Machine Learning Databases*. Accessed: Nov. 28, 2017. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[46] L. Breiman, "Random forest," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[47] T. Roos, H. Wettig, P. Grünwald, P. Myllymäki, and H. Tirri, "On discriminative Bayesian network classifiers and logistic regression," *Mach. Learn.*, vol. 59, no. 3, pp. 267–296, 2005.

[48] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proc. 13th IJCAI*, Chambery, France, 1993, pp. 1022–1029.
[49] B. Cestnik, "Estimating probabilities: A crucial task in machine learning," in *Proc. 9th ECAI*, Stockholm, Sweden, 1990, pp. 147–149.
[50] Z. Duan and L. Wang, "$K$-dependence Bayesian classifier ensemble," *Entropy*, vol. 19, no. 12, p. 651, Nov. 2017.
[51] J. H. Xue and D. M. Titterington, "Comment on 'On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes,'" *Neural Process. Lett.*, vol. 28, no. 3, pp. 169–187, Dec. 2008.
[52] P. Hadjicostas, "Consistency of logistic regression coefficient estimates calculated from a training sample," *Statist. Probab. Lett.*, vol. 62, no. 3, pp. 293–303, Apr. 2003.
[53] A. Genkin, D. D. Lewis, and D. Madigan, "Large-scale Bayesian logistic regression for text categorization," *Technometrics*, vol. 49, no. 3, pp. 291–304, Aug. 2007.
[54] L. Wang and H. Zhao, "Learning a flexible k-dependence Bayesian classifier from the chain rule of joint probability distribution," *Entropy*, vol. 17, no. 6, pp. 3766–3786, Nov. 2015.
[55] R. Kohavi and D. Wolpert , "Bias plus variance decomposition for zero-one loss functions," in *Proc. 13th ICML*, Bari, Italy, 1996, pp. 275–283.
[56] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochim. Biophys. Acta-Protein Struct.*, vol. 405, no. 2, pp. 442–451, Oct. 1975.
[57] J. Gorodkin, "Comparing two $K$-category assignments by a $K$-category correlation coefficient," *Comput. Biol. Chem.*, vol. 28, nos. 5–6, pp. 367–374, Dec. 2004.
[58] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Statist. Assoc.*, vol. 32, no. 200, pp. 675–701, Dec. 1937.
[59] P. Nemenyi, "Distribution-free multiple comparisons," Ph.D. dissertation, Downstate Med. Center, Princeton Univ., Princeton, NJ, USA, 1963.
[60] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.



**ZHIYI DUAN** received the B.Eng. degree from Jilin University, Changchun, China, in 2014, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology. His research interests include data mining and Bayesian networks.

**LIMIN WANG** received the Ph.D. degree in computer science from Jilin University, China, in 2005, where he is currently a Professor with the College of Computer Science and Technology. He has published innovative papers in journals, such as *Knowledge-Based Systems*, *Expert System with Applications*, and *Progress in Natural Science*. His research interests include probabilistic logic inference and Bayesian networks.

**MINGHUI SUN** received the Ph.D. degree in computer science from the Kochi University of Technology, Japan, in 2011. He is currently an Assistant Professor with the College of Computer Science and Technology, Jilin University, China. He is interested in using HCI methods to solve challenging real world computing problems in many areas, including tactile interface, pen-based interface, and tangible interface.